**Containers**

* name: the identifier of a container; a random id will be given if absent
* image: in the form [registry]/image[:tag]; hyper will automatically pull the image if missing
* command: the shell command to run when the container starts
* entryPoint: the executable to run when the container starts
  + The command will be appended to entryPoint as parameters if entryPoint is not empty.
* workdir: the directory running the container command
* ports: the exposed ports of the container
  + containerPort: the listening port inside container
  + hostPort: the port exposed in host machine
  + protocol: tcp (default) or udp
* volumes: volumes to be mounted in the container.
  + path: the mount point
  + volume: the name of the volume to be mounted, defined in [volumes](https://docs.hyper.sh/reference/volumes.html) section
  + readOnly: if true, the mount point will be read only, default false
* files: files to be present in the container
  + path: the file path in the container
  + filename: the filename defined in [files](https://docs.hyper.sh/reference/files.html) section
  + perm: the file permission, by default 0755
* restartPolicy: restart the container if exit: never, onFailure, or always

example:

"containers" : [{

"id": "app",

"image": "repo/image:tag",

"command": ["/bin/sh"],

"workdir": "/root",

"envs": [{

"env": "JAVA\_OPT",

"value": "-XMx=256m"

}],

"volumes": [{

"path": "/var/log",

"volume": "name",

"readOnly": false

}],

"files": [{

"path": "/var/lib/xxx/xxxx",

"filename": "name",

"perm": "0755"

}],

"ports":[{

"containerPort": 80,

"hostPort": 8000,

}],

"restartPolicy": "never"

}]